aws

COMMUNITY DAY

CAMEROON

22nd November 2025, Buea

BUILDING A SERVERLESS DATA PIPELINE ON AWS:

A Case Study from Cameroon

TENGU Quinter

AWS Solution Architect Instructor @ Trustech Online University

Founder & CEO: Elegancia Cakes & Treats

My Socials

LinkedIn: Tengu Quinter

Facebook: Quinter T Munah

Facebook: Elegancia Cakes and Treats

IG: Royal Elegancia Cakes

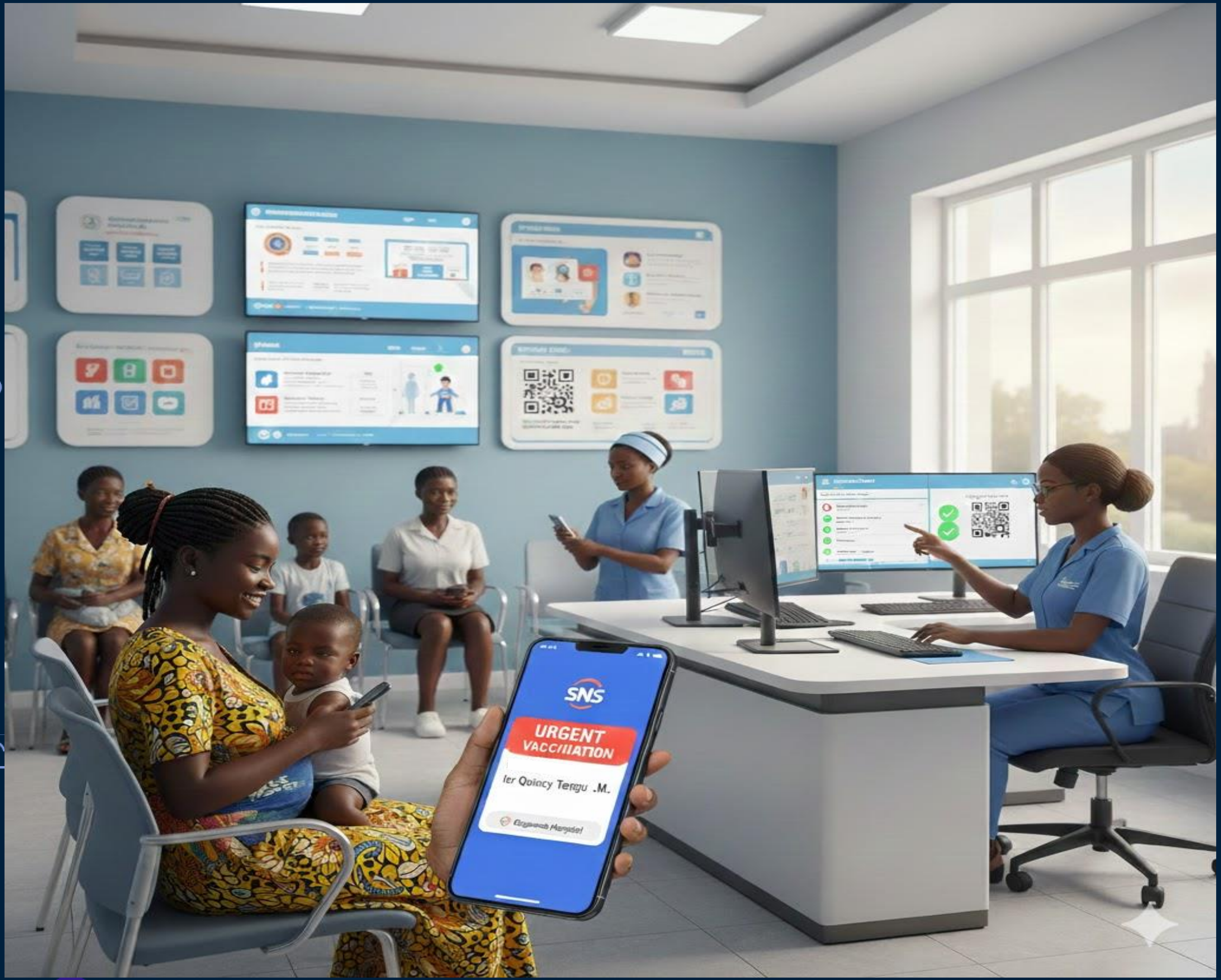
As a mom of 4, I'm highly aware of how easy it is to miss critical vaccination dates and I’m guilty of that.

We still have to get it.

The Challenge

Manual tracking of health records including vaccination records are prone to errors which can have life altering consequences.

Problem Solved

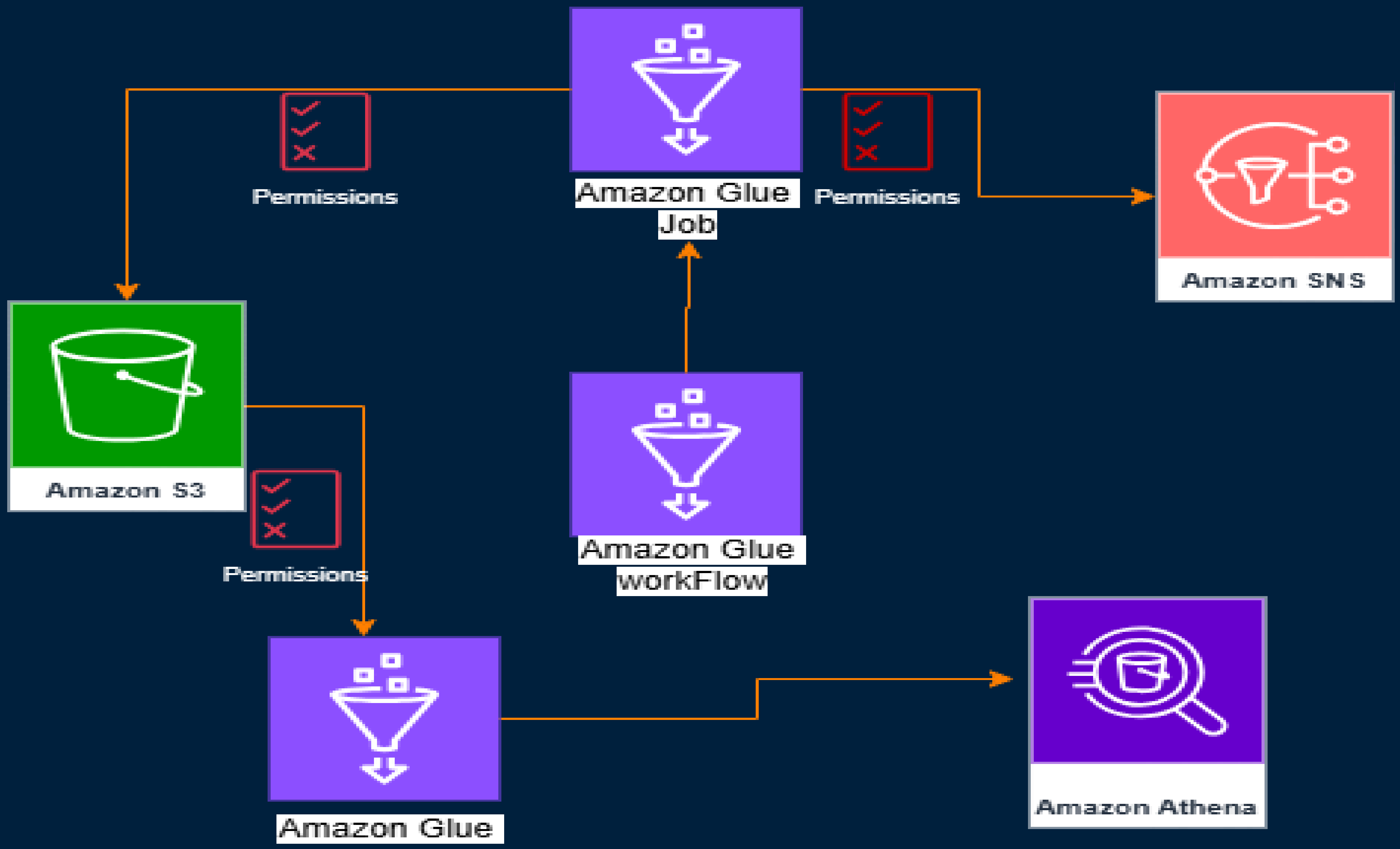


A Serverless Data Pipeline digitizes patient schedule, thereby replacing manual checks and ensuring proactive, timely, life-saving vaccination.

Problem Solved

A Serverless Data Pipeline digitizes patient schedule, thereby replacing manual checks and ensuring proactive, timely, life-saving vaccination.

A Serverless Data Pipeline on AWS



How It Works: the Pipeline Flow

1. **Data Ingestion**: A healthcare worker uploads a CSV file containing patient data to an S3 bucket.
2. **Automated Processing**: A **Glue Workflow** triggers a **Glue Job** on a daily schedule.
3. **ETL**: The Glue Job processes the data, identifying babies with near due vaccinations.

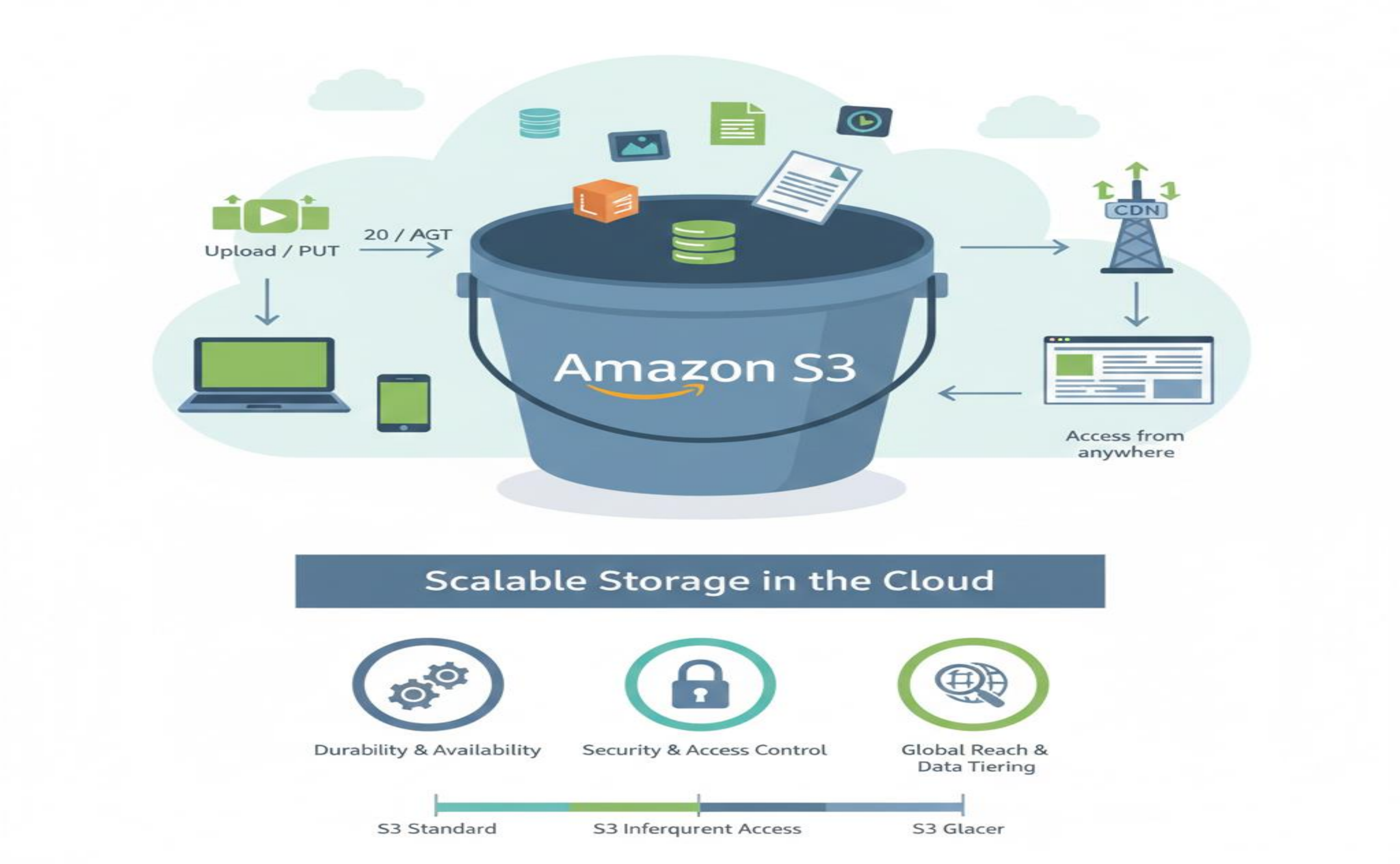
4. **Proactive Notifications**: The Glue Job publishes two separate messages to an **SNS topic**, one for babies ready for vaccination and another if there are no babies due for vaccination.

How It Works: the Pipeline Flow

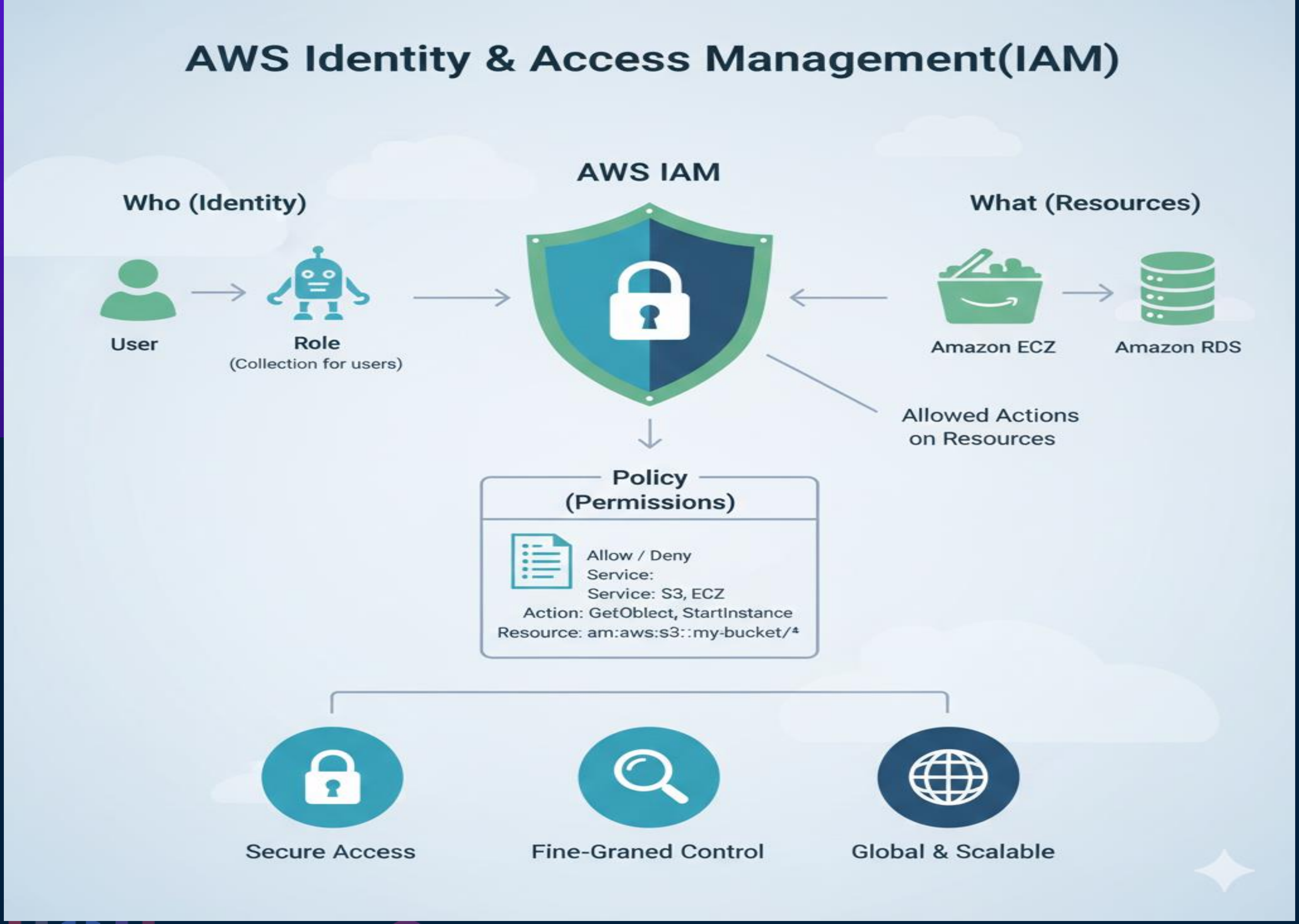
5. Email Reminders: SNS delivers these messages to the subscribed Public Relations Officer (PRO) via email.

6. Data Analysis: The PRO or other stakeholders can use **Athena** to run SQL queries or create a dashboard in **QuickSight** to analyze vaccination trends and forecasts.

AWS Services: A Deep Dive


Upload / PUT
20 / AGT
CDN
Amazon S3
Access from anywhere
Scalable Storage in the Cloud
Durability & Availability
Security & Access Control
Global Reach & Data Tiering
S3 Standard
S3 Inferqurent Access
S3 Glacer

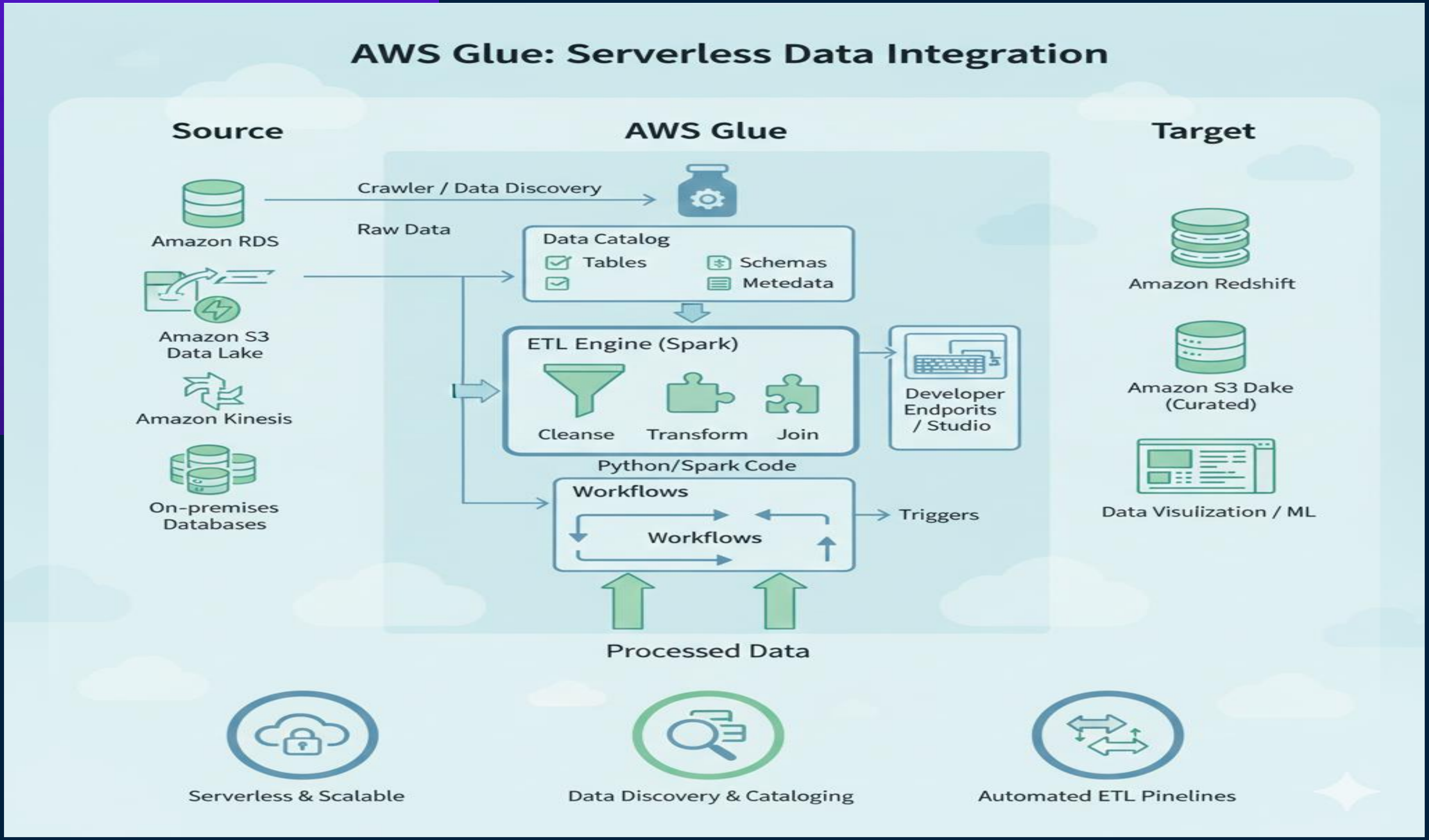
AWS Services: A Deep Dive



AWS Identity & Access Management(IAM)

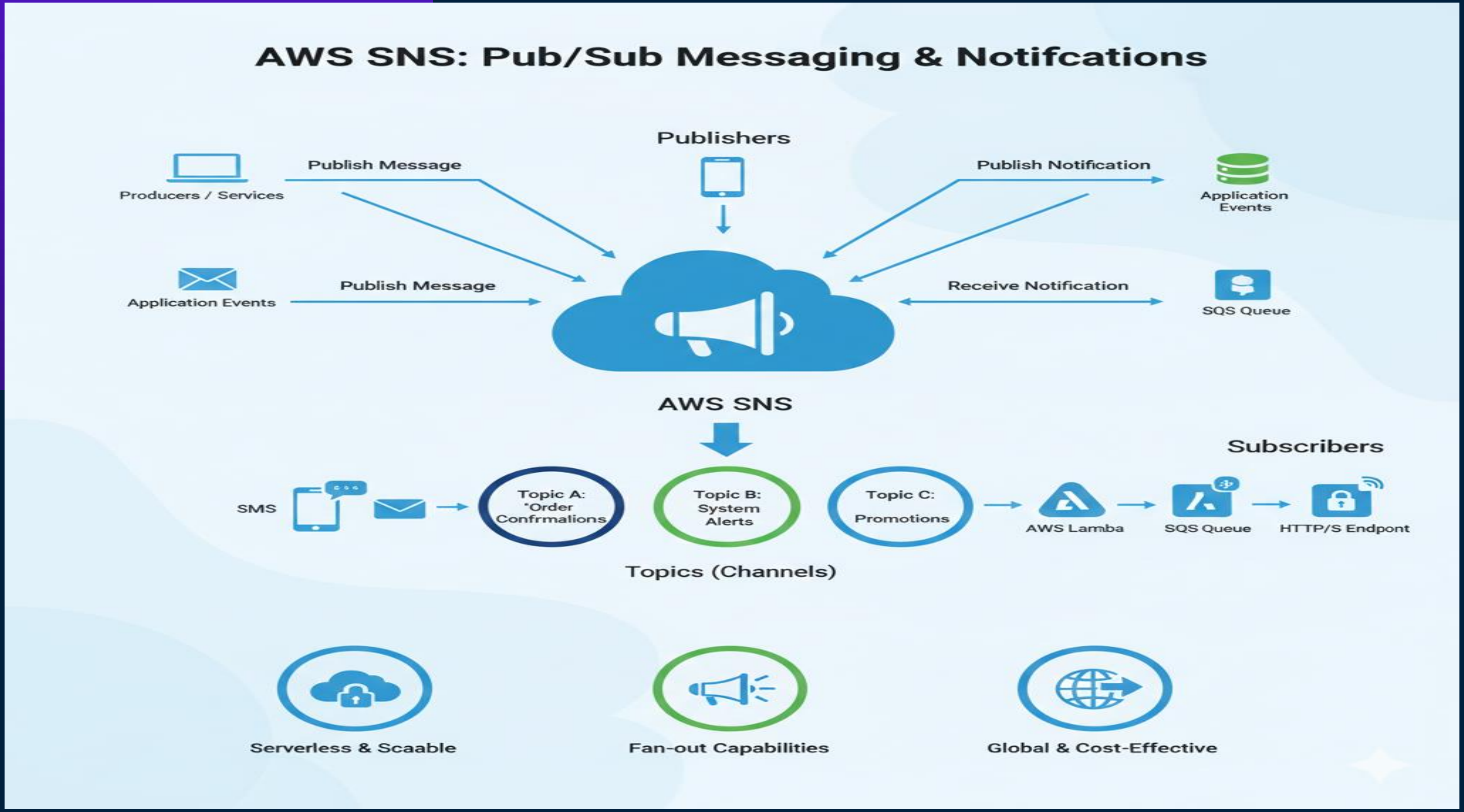
Allow / Deny
Service:
Service: S3, ECZ
Action: GetOblect, StartInstance
Resource: am:aws:s3::my-bucket/*

AWS Services: A Deep Dive



AWS Glue: Serverless Data Integration

AWS Services: A Deep Dive



SETUP AND INSTRUCTIONS

STEP 1: DATA AND STORAGE 📁

Preparing Our Dataset

- **Create an S3 Bucket:**
  - Go to **Amazon S3** and create a new bucket.
  - Name it descriptively (e.g., vaccination-data-2025).

- **Upload the File:**
  - Upload the vaccination records gotten from the hospital to the S3 bucket.
  - Name it descriptively (e.g., vaccination-records-2025).

STEP 2: NOTIFICATIONS 🔔

Setting up a Channel for Reminders

- **Create an SNS Topic:**
  - Go to **Amazon SNS** and click **Create topic**.
  - **Name:** pro-vaccination-reminders.
  - **Type:** Standard.

- **Add an Email Subscription:**
  - Within the topic, click **Create subscription**.
  - **Protocol:** Email.
  - **Endpoint:** pro@example.com (add PRO’s email).

- **Confirm Subscription:** The PRO must check their email
- and click the confirmation link.

STEP 3: THE GLUE JOB 💻

The Engine of Our Pipeline

- **Create IAM Role: Glue Role**
  - Attach policies for S3, SNS, and CloudWatch access.
  - Attach the AWSGlueServiceRole managed policy.
- **Modify the Glue script:**
- **C**opy your **S3 Bucket Name,Key(csv file name)** and **SNS Topic ARN** and paste on the Glue script
- **Create the Glue Job:**
  - Go to **AWS Glue** > **Jobs**.
  - Select **Script editor.**
  - **Type: Spark** (critical for pyspark).
  - Uplaod script from your pc and create script.

STEP 3: THE GLUE JOB 💻

- Go to **Job details** and entire the details.
- **Name:** vaccination-reminder-job.
- **IAM Role:** Select the role you created.

Language: python 3

- **Script Path:** s3://your-bucket/glue_script.py.
- **Worker Type:** G.1X.
- **Workers:** 2. we don’t need the default 10 workers for this job.
- Leave other default configurations and save.
- You can test the job by running it manually.

STEP 4: THE TRIGGER □

The Daily Scheduler

- **Create Glue Workflow:**
  - Go to **AWS Glue** > **Workflows**.
  - Click **Add workflow**.
  - Name: vaccination-reminder-workflow.
- **Create the Trigger:**
  - Add a trigger to your workflow.
  - **Schedule Type:** Scheduled.
  - **Frequency:** choose **Custom** and add rate(5 minutes) OR Cron Expression: 0/5 * * * ? *.
  - **Add Node:** Connect your vaccination-reminder-job to this trigger.
- **Start the Workflow:** The workflow will now automatically execute the job every 5 minutes. In real life it will be daily.

STEP 5: DATA ANALYSIS WITH ATHENA 📊

Insights from Our Data

- **Create a Glue Crawler:**
  - Go to **AWS Glue** > **Crawlers**.
  - Point it to your S3 bucket.
  - Create an IAM Role add GlueServiceRole as well as S3BucketFullAccess
  - Create the database vaccination-Reminder-db.
  - Run the crawler to create the table vaccination_records.

STEP 5: DATA ANALYSIS WITH ATHENA 📊

- **Query Your Data:**
  - Open the Table your crawler created,
  - Click on Action and click view data
  - Click Proceed
  - Go to settings and add an S3 bucket as your query result location.

Run queries to get insights, like finding specific babies to view their records.

```
SELECT
    * FROM "AwsDataCatalog"."etl-
db"."vaccination_data_2025"
WHERE
    "col0" IN ('Agnes Beye', 'Bella Mballa', 'Samir Ngono');
```

TROUBLESHOOTING 💡

Common Issues and Solutions

- **ModuleNotFoundError: No module named 'pyspark'**
  - **Fix:** Ensure your Glue job **Type** is **Spark**, not Python Shell.
- **TABLE_NOT_FOUND**
  - **Fix 1:** Verify your Glue crawler is pointing to the correct S3 path (e.g., s3://bucket/, not s3://bucket/file.csv).
  - **Fix 2:** In Athena, ensure the database dropdown is set to the correct database (e.g., vaccine-db2).
  - **Fix 3:** Check for typos in the database or table name.

Delete all resources you created to save cost.

THANK YOU